

AT END OF PAGE

Structured Mode Syntax

```
[AT] END [OF] PAGE [(rep)]  
    statement...  
END-ENDPAGE
```

Reporting Mode Syntax

```
[AT] END [OF] PAGE [(rep)]  
    { statement  
      DO statement... DOEND }  
}
```

Related Statements: AT TOP OF PAGE | DISPLAY | WRITE | INPUT | NEWPAGE

Function

The AT END OF PAGE statement is used to specify processing that is to be performed when an end-of-page condition is detected (see the session parameter PS in the Natural Parameter Reference documentation).

An end-of-page condition may also occur as a result of a SKIP or NEWPAGE statement, but not as a result of an EJECT or INPUT statement.

An AT END OF PAGE statement block is only executed if the object which contains the statement block is active at the time when the end-of-page condition occurs.

An AT END OF PAGE statement must not be placed within an inline subroutine.

This statement is non-procedural (that is, its execution depends on an event, not on where in a program it is located).

Report Specification - rep

The notation (*rep*) may be used to specify the identification of the report for which the AT END OF PAGE statement is applicable. A value in the range 0 - 31 or a logical name which has been assigned using the DEFINE PRINTER statement may be specified.

If (*rep*) is not specified, the AT END OF PAGE statement will apply to the first report (Report 0).

Logical Page Size

The end-of-page check is performed after the processing of a DISPLAY or WRITE statement is completed. Therefore, if a DISPLAY or WRITE statement produces multiple lines of output, overflow of the physical page may occur before an end-of-page condition is detected.

A logical page size (session parameter PS) which is less than the physical page size must be specified to ensure that information printed by an AT END OF PAGE statement appears on the same physical page as the title.

Last-Page Handling

Within a main program, an end-of-page condition is activated when the execution of the main program terminates via ESCAPE, STOP or END.

Within a subroutine, an end-of-page condition is not activated when the execution of the subroutine terminates via ESCAPE, RETURN or END.

System Functions

Natural system functions may be used in conjunction with an AT END OF PAGE statement as described in the section Using System Functions in Processing Loops of the Natural System Functions documentation.

If a system function is to be used within an AT END OF PAGE statement block, the GIVE SYSTEM FUNCTIONS clause must be specified in the corresponding DISPLAY statement.

INPUT Statement with AT END OF PAGE

If an INPUT statement is specified within an AT END OF PAGE statement block, no new page operation is performed. The page size (session parameter PS) must be reduced to a value that allows the lines created by the INPUT statement to appear on the same physical page. See also INPUT statement "Split Screen Feature". See also Example 2.

Example 1

```

/* EXAMPLE 'AEPEX1S': AT END OF PAGE (STRUCTURED MODE)
/*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
  2 JOB-TITLE
  2 SALARY (1)
  2 CURR-CODE (1)
END-DEFINE
/*****
FORMAT PS=10
LIMIT 10
READ EMPLOY-VIEW BY PERSONNEL-ID FROM '20017000'
  DISPLAY NOTITLE GIVE SYSTEM FUNCTIONS
    NAME JOB-TITLE 'SALARY' SALARY(1) CURR-CODE (1)
/* ****
  AT END OF PAGE
    WRITE / 28T 'AVERAGE SALARY: ...' AVER(SALARY(1)) CURR-CODE (1)
  END-ENDPAGE
/* ****
END-READ
/*****
END

```

NAME	CURRENT POSITION	SALARY	CURRENCY CODE
-----	-----	-----	-----
CREMER	ANALYST	34000	USD
MARKUSH	TRAINEE	22000	USD
GEE	MANAGER	39500	USD
KUNEY	DBA	40200	USD
NEEDHAM	PROGRAMMER	32500	USD
JACKSON	PROGRAMMER	33000	USD
AVERAGE SALARY: ...		33533	USD

Equivalent reporting-mode example: See program AEPEX1R in library SYSEXRM.

Example 2

```

/* EXAMPLE 'AEPEX2': AT END OF PAGE WITH INPUT STATEMENT
/*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
2 NAME
2 FIRST-NAME
2 POST-CODE
2 CITY
1 #START-NAME(A20)
END-DEFINE
/*****
FORMAT PS=21
REPEAT
  READ (15) EMPLOY-VIEW BY NAME = #START-NAME
  DISPLAY NOTITLE NAME FIRST-NAME POST-CODE CITY
END-READ
NEWPAGE
/*****
  AT END OF PAGE
  MOVE NAME TO #START-NAME
  INPUT / '-' (79) / 10T 'Reposition to name ==>'
    #START-NAME (AD=MI) '('.'.' to exit)'
  IF #START-NAME = '.'
    STOP
  END-IF
  END-ENDPAGE
/*****
END-REPEAT
END

```

NAME	FIRST-NAME	POSTAL ADDRESS	CITY
-----	-----	-----	-----
ALEXANDER	STEPHEN	19711	NEWARK
ALEXANDER	GIL	21209	BALTIMORE
ALEXANDER	CHARLY	95616	DAVIS
ALEXANDER	HOLLY	53706	MADISON
ALLEGRE	KARL-OTTO	8100	CHARLEVILLE MEZIERES
ALLSOP	ALAN	DE3 3NL	DERBY
ALVAREZ	RAQUEL	28015	MADRID
AMOROS	FUENSANTA	28014	MADRID
ANDERSEN	ANITA	1850 V	KÖBENHAVN
ANDERSEN	KARIN	2720	VANLÖSE
ANDERSEN	LISSI	2650	HVIDOVRE
ANDERSON	JENNY	84112	SALT LAKE CITY
ANTLIFF	JANET	DE3 3EE	DERBY
ARCHER	ROBIN	DE4 8GR	DERBY
ARCONADA	ARANZAZU	28014	MADRID
-----	-----	-----	-----
Reposition to name ==> ARCONADA		('.' to exit)	

